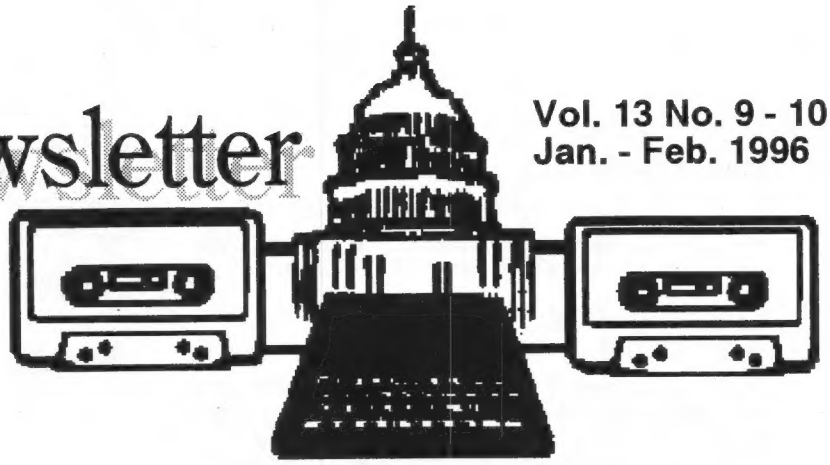
CATS Newsletter

The Capital Area Timex/Sinclair Users Group

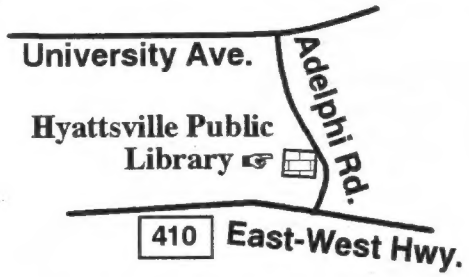
Vol. 13 No. 9 - 10
Jan. - Feb. 1996



CATS Meeting Site Moves!

CATS will be meeting at the Hyattsville branch of the public library system from February to June, due to unreported sliding renewal deadlines and eager Saturday tutoring programs at New Carrollton. A tag-team scheduling committee of Joe Miller, Mark Fisher, and Barry Washington handled recovery from the fumble.

The library is located at 6530 Adelphi Rd. To reach the library: 1) Exit southbound from Washington Beltway on Hwy 1. 2) Turn right on route 410 (East-West Hwy.) 3) Turn right on Adelphi Rd. (6th traffic light) 4)As you come even with the library in one block, turn left on Toledo Rd. 5) Library is on left. 6) The meeting room is the first door on the left on entering the library. 7) (Note: bring quarters if parking on Toledo road). The library phone # is (301) 779-9330.The map below shows its location:



July through December will again be at New Carrollton.

From QL-Hacker's Journal: #23, January 1996

While surfing on the Net (as the media likes to hype) I ran into a site that had some essays on programming. One, "Notes on Programming in C" by Robert Pike had some interesting points. TS. [So what was the URL Tim?]

NOTES ON PROGRAMMING IN C

by **Robert Pike**

Procedure Names

Procedure names should reflect what they do; function names should reflect what they return. Functions are used in expressions, often in things like IF's, so they need to read appropriately.

IF (CHECKSIZE(x))

is unhelpful because we can't deduce whether CHECKSIZE returns true on error or non-error, instead

IF (VALIDSIZE(x))

makes the point clear and makes a future mistake in using the routine less likely.

Complexity

Fancy algorithms are buggier than simple ones, and they're much harder to implement. Use simple algorithms as well as simple data structures.

Continued on page 2...

From the Editor:

We're on for February!
Between snow and scheduling, January was a bust. But, thanks to Barry Washington and Joe Miller, we've got the next year's meetings scheduled.

February will be the next Paste-up Month. Bring the articles you'd like to see in the newsletter and I'll pick them up to include in the newsletter. Or post them to the club address on the internet (see back page).

Do we have your internet address? I propose setting up a "CATS mailing list" to distribute comments of interest to the members.

Our current list is as follows:

72047.2774@compuserve.com, ROBBINS4037@delphi.com, hlschaaf@UDel.Edu, jw0044, mf0002, rd0007, swensont@mail.serve.com

From page 1...

Data dominates

If you've chosen the right data structures and organized well, the algorithms will almost be self-evident. Data structures, not algorithms, are central to programming.

Programming with Data

Algorithms, or details of algorithms, can often be encoded compactly, efficiently and expressively as data rather than, say, as lots of IF statements. The reason is that the complexity of the job at hand, if it is due to a combination of independent details, can be encoded.

A classic example of this is parsing tables, which encode the grammar of a programming language in a form interpretable by a fixed, fairly simple piece of code. Finite state machines are particularly amenable to this form of attack, but almost any program that involves the 'parsing' of some abstract sort of input into a sequence of some independent 'actions' can be coned profitably as a data-driven algorithm.

One of the reasons data-driven programs are not common, at least among beginners, is the tyranny of Pascal. Pascal, like its creator, believes firmly in the separation of code and data. It therefore (at least in its original form) has no ability to create initialized data. This flies in the face of the theories of Turing and von Neumann, which define the basic principles of the stored-program computer. Code and data are the same, or at least they can be. How else can you explain how a compiler works?

Function Pointers

Another result of the tyranny of Pascal is that beginners don't use function pointers. (You can't have function-valued variables in Pascal.) Using function - pointers to encode complexity has some interesting properties.

Some of the complexity is passed to the routine pointed to. The routine must obey some standard protocol - it's one of a set of routines invoked identically - but beyond that, what it does is its business alone. The complexity is distributed.

There is this idea of a protocol, in that all functions used similarly must behave similarly. This makes for easy documentation, testing, growth and even making the program run distributed over a network - the protocol can be encoded as remote procedure calls.

I argue that clear use of function pointers is the hard of object-oriented programming. Given a set of operations you want to perform on data, and a set of data types you want to respond to those operations, the easiest way to put the program together is with a group of function pointers for each type. This, in a nutshell, defines class and method. The O-O languages give you more of course, - prettier syntax, derives types and so on - but conceptually they provide little extra.

Combining data-driven programs with function pointers leads to an astonishingly expressive way of working, a way that, in my experience, has often lead to pleasant surprises. Even without a special O-O language, you can get 90% of the benefit for no extra work and be more in control of the result. I cannot recommend an implementation style more highly. All the programs I have organized this way have survived comfortably after much development - far better than with less disciplined approaches. Maybe that's it: the discipline it forces pays off handsomely in the long run.

Bosnia Endeavor Goes on the Web

by Robert Thomason

From Dec. 8, 1995 Washington Post. Contributed by Barry Washington. The Defense Department has found a new way to keep the home front in touch with its Bosnia operation. The Pentagon has established BosniaLINK, a WWW site that provides the public basic information about U.S. troops in the Balkans. The site links the viewer to dozens of other DOD documents on Operation Joint Endeavor.

During its first week, BosniaLINK displayed maps of the American, French, and British sectors, provided bio's of U.S. high command, and described 14 U.S. Army units deployed to the area. It even listed the standard issue of field equipment and clothing for cold climates.

The Pentagon hopes to add pictures taken by the 55th Signal Co. from Fort Meade, MD. At this point, BosniaLINK users will not be able to send messages to the troops in Bosnia. The address for BosniaLINK is www.dtic.dla.mil/bosnia

DATABASES AND BITMAPS

Tim Swenson

From QL-Hacker's Journal: #23, January 1996

I read an interesting article that introduced the idea of using bitmaps in implementing a database. I don't know if this concept will be of use to any QL programmers, but I found it interesting just to know conventional databases use B-trees and hashing to implement indexes. B-trees are just that, trees. An index is created in a tree structure lumping like records together. Hashing is using a mathematical formula to distribute the records into an array, so that when you need to find them again, you just plug the record into the formula and it takes you to the proper place in the array.

These structures work well in transaction-oriented systems (find, edit, delete), but they begin to have problems when the number of conditions of a query increase. The more ANDs and ORs the longer these structures take to search. Enter the bitmap index. Bitmap indexes store all possible values for a field into a bitmap. Bitmaps can be ANDed and ORed quickly to return a result. Queries can be made with many conditions in the index without even looking at the actual data in the records.

Here is an example of a bitmap index. Using the illustration below, note that each record (1-7) has a 4-bit

index which is holds the color of the car. With four possible values for color, the appropriate bit is turned on for each color.

	1 2 3 4 5 6 7	
Blue	1 0 0 0 0 1 0	Cars 1 and 6 are Blue.
Orange	0 1 0 0 0 0 0	Car 2 is Orange.
Green	0 0 0 0 1 0 1	Cars 5 and 7 are green.
Red	0 0 1 1 0 0 0	Cars 3 and 4 are red

Below is the query for "which cars are red, made in 1995, and cost $19,000." Each record has a bitmap index for each of the values. To get the result the three bitmaps are ANDed and the result shows in the Result column: Car #4. ORing three 7 bit numbers can be done very quickly.

	Red	1995	$19K	Result
1	0	0	0	0
2	0	0	1	0
3	1	0	0	0
4	1 and	1 and	1 =	1
5	0	0	0	0
6	0	0	0	0
7	0	1	0	0

Bitmap indices have two major problems: they are hard to update and can't handle high cardinality. Cardinality is the number of different values each field can have. If you have a database of cities, there are many possible values for the city. Where as in this car example, there are only so many different colors a car can be. Bitmaps can take substantially longer to update than a plain B-tree.

Like with most computer solutions, the most optimal solution may be a combination of all structures. Using B-trees, Hashing, and Bitmap indexes, in conjunction, may work out to be the best approach.

[Problems we don't have:]

Users of Microsoft Word susceptible to new virus

Stephen Manes

Personal Computers, The Sun, 9/18/95

Contributed by **Ruth Fegley**

A NEW KIND of computer virus has descended upon the world. How easy is it to create one?

Fifteen minutes after opening a Microsoft Word reference manual, I had cranked out a one-line program that could delete crucial system files from a hard drive. After an hour I had adapted the program to run automatically whenever anyone opened a file called HELPFUL.DOC.

By bedtime I had figured out how to get this file to transmogrify Word itself so that it would embed my trick program in any document it opened. In an evening, I had created a virus of my very own. This is scary stuff. Lacking the skill and knowledge to create a classic computer virus, I managed to put together this new kind with relative ease. Scarier still is that if I can do it, millions of others can too.

I Worse, to wreak its havoc, this type of virus requires only that its recipient open an innocent-looking document. Conventional wisdom in the networked world had suggested that one could elude viruses by shunning program files of unknown provenance. No longer

The realm of computer viruses has suddenly grown larger, though all that has really changed is the advent of a pointedly devilish example from points unknown. The Microsoft Corp. prefers to call it a "prank macro"; others have dubbed it the "Word concept virus." Macros are simple commands that do complicated things. A rudimentary macro, for example, might record the word "Festschrift" as you typed it and then play it back every time you pressed Alt-F.

But modem macros are full-blown programs that do everything from creating standard forms to changing the way menus work. Workers often use macros without knowing it. Languages for writing macros in today's applications offer so much power that programmers can do almost anything with them.

So, too, can saboteurs. Although these languages have been around for years, the prank virus stunningly reveals their tremendous potential for trouble. Fortunately the prank itself is mostly harmless. Use Microsoft Word to open a document containing it, and the virus infects Word so that it saves files only in so-called template form.

That matters hardly at all, except that every time someone else loads one of those templates, it will infect that copy of Word so that it can infect other files. That no data are truly harmed suggests kindness on the part of the unknown author, who includes the phrase "That's enough to prove my point" in the body of the virus.

Since discovering the problem, the Microsoft Corp. has spread the word about it. The company also inadvertently spread the virus via an infected file called OEMLTR.DOC on a CD-ROM called the "Windows System Compatibility Test," which most users are unlikely to encounter.

But Microsoft spokesmen say no copies of Word itself have been sold with the virus code.

The macro can infect Word 95 and Word 6 documents. Unlike classic viruses, which are highly specific, to a particular type of machine and operating system, this kind can work wherever you run Microsoft Word, including PCs using Windows 3.1, 95 and NT, as well as Macintoshes.

To check for it, pull down Word's Tools menu and click Macros. If items called AAZAAO and AAAZFS appear in the list box, your copy of Word and a document are infected.

New York Times News Service: Stephen Manes and Peter H. Lewis alternate writing this column, which appears every Monday.

[Ed. note: Now Ruth, how about sending in a happy story!]

THE CAPITAL AREA TIMEX/ SINCLAIR USERS GROUP is a not-for-profit group serving the interests of those who own, use, or are interested in the Timex/Sinclair family of computers.

Officers:

Chairman	Monthly position
Vice-Chairman	Monthly position
Recording Secretary	Monthly position
Corresponding Secretary	Joe Miller
Treasurer	Barry Washington
Newsletter Editor	Mark Fisher

Voice, FAX, and DATA for articles: (301) 589-7407

Meetings:

Monthly meetings are held from 12:00 M to 4:30 PM on the second Saturday of each month at the Hyattsville Library.

Membership:

User Group memberships cost $10.00 per year, are good for 12 months and include all privileges (access to software libraries, group buys, etc.).

Mail dues to:

LeBaron (Barry) Washington
7044 Cindy Lane
Annandale VA 22003

Networks

internet: send article submissions (short or long) to our club address, below.

Timex SIG on Compuserve: Wednesday night 10:00 PM Eastern Standard Time (GO CLUB)

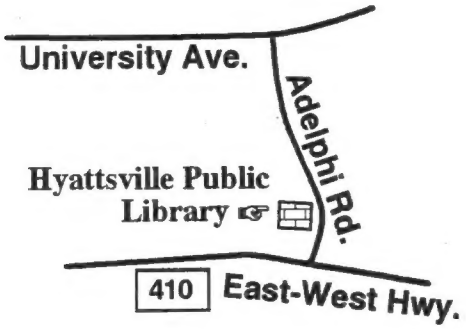
QBOX BBS: (810) 254-9878

CATS on Internet:

mf0002@epfl2.epflbalto.org

CATS Newsletter
7044 Cindy Lane
Annandale VA 22003

FIRST CLASS MAIL



LAMBERT, DONALD S. 06/96
1301 KIBLINGER PLACE
AUBURN, IN 46706-3010

The next two meetings of CATS will be held on: February 10

(Paste up for March - April)

and ... March 9, 1996

12:00 N Hardware workshop
2:00 PM General meeting

At: Hyattsville Public Library